



## King's Research Portal

*Document Version*  
Peer reviewed version

[Link to publication record in King's Research Portal](#)

*Citation for published version (APA):*

Alaboud, F. K. M., & Coles, A. I. (2018). Personalised Medication Planning using PDDL+. In S. Bernadini, S. Parkinson, & K. Talamadupula (Eds.), *Proceedings of the Scheduling and Planning Applications woRKshop (SPARK): At the 28th International Conference on Automated Planning and Scheduling*

### **Citing this paper**

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

### **General rights**

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

### **Take down policy**

If you believe that this document breaches copyright please contact [librarypure@kcl.ac.uk](mailto:librarypure@kcl.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

# Personalised Medication Planning using PDDL+

Fares K. Alaboud and Andrew Coles

Department of Informatics, King's College London, UK

email: `firstname.lastname@kcl.ac.uk`

## Abstract

Prescription medication is typically prescribed with a standardised set of instructions, to be followed regularly, with the aim being to manage symptoms while remaining within safe dosage limits. The caveat of such standardisation is that it is not tailored to the needs of the patient, in terms of their activities. In this paper, we take the first steps towards modelling medication pharmacokinetics as a PDDL+ hybrid planning problem. As pharmacokinetics are inherently non-linear, we present a planner-independent linearise-validate cycle, where tasks can be solved by iterative refinement of a linear approximation of the domain, by validation against the full non-linear semantics.

## 1 Introduction

One of the largest problems in healthcare is the incorrect consumption of medication. It is estimated that half of patients that are prescribed medication for chronic conditions do not consume their medication correctly (The Academy of Medical Sciences 2014). Most medication is prescribed in a way that expects the patient to follow a standard routine. This is done in order to help the patient stay compliant and at the same time to consume the medication in a way that does not endanger the patient – often, when patients are given a regular dose, it is to keep things simple. For example, paracetamol (acetaminophen) is usually given in doses of 500mg per pill. The standard dose is two pills to be taken every four to six hours, with a maximum consumption of eight pills per day. Higher levels may give more pain relief, but the rate at which it is metabolised gives a risk of paracetamol toxicity if these limits are exceeded – the spacing between doses, and daily limit, avoid excess exposure.

To address the challenge of effectively managing patients' medication usage, one option is to produce personalised medication plans. Personalised medicine is defined by the as providing “the right patient with the right drug at the right dose at the right time” (Sadée and Dai 2005). Historically, the scope for this has been limited to where it is essential (for instance, personalised insulin regimes for diabetics) but is recently becoming more viable through the uptake of technology – at one extreme, with the use of a drug dosage printer to ‘print’ drugs with accurately specific doses (Hirshfield et al. 2014).

In this paper we present the possibility of using PDDL+ to personalise medication schedules by modelling the problem as a hybrid planning domain, determining an effective schedule for a patient depending on their varying pain relief needs throughout the day. As the metabolism of medication is non-linear (negative exponential), and many otherwise-effective PDDL planners do not support non-linear domains, we explore the use of an iterative piece-wise linear approximation process to allow a broader range of planners to be used as a kernel within this process; and hence find solutions that are valid when considering the full non-linear pharmacokinetics. We present an initial evaluation of this approach using the planner OPTIC (Benton, Coles, and Coles 2012), as extended to support PDDL+ (Coles and Coles 2014), and discuss the future direction of the work and limitations of PDDL+ for modelling desirable objective functions in this domain.

## 2 Background

When consumed, medication is metabolised in the body over time, leading to a decay of the active medication level. Whilst pharmacokinetics are complex, a reasonable model is to assume negative-exponential (i.e. first-order) decay, with drug-dependent half-lives depending on the rate at which the active ingredients are metabolised (Geenen et al. 2013). Returning to the example of Paracetamol, the half life is up to 3 hours. That means if someone takes 1000mg of paracetamol at 12.00pm, there will be 500mg of the drug left in three hours (i.e. at 3.00pm). In another three hours (i.e. 6.00pm), there will be 250mg of the drug left, and so on. The question then, returning to the topic of this paper, is how these pharmacokinetics can be modelled in PDDL.

Medication levels changing over time are an example of *continuous numeric change*. The capability for these was first added to PDDL in PDDL2.1 (Fox and Long 2003), as part of ‘layer 5’ – durative actions can have continuous numeric effects that occur during their execution. As drug metabolism is not something that one can *choose* to occur in a plan, but is rather something that occurs exogenously in the world, a better fit is to encode it as a process, expressed in PDDL+ (Fox and Long 2006). PDDL+ provides a language for defining hybrid planning problems, where the state trajectory for a solution plan contains the effects of the planned actions (as in PDDL2.1) but also the consequences of ex-

ogenous processes and events. The key distinction is that a process occurs *whenever* its conditions are true: it stops and starts, outwith the direct control of the planned actions; and during its execution, it effects continuous numeric change upon the world. Analogously, PDDL+ events are instantaneous actions that occur *whenever* their conditions are true: when they fire, the world is immediately updated according to their instantaneous effects. By combining these, PDDL+ provides a useful toolkit for expressing hybrid domain models for a range of problems (Piotrowski et al. 2015). In this paper, as we will illustrate, we combine these to provide the necessary exogenous context in which to plan personalise medication consumption schedules.

### 3 High-level Problem Description

The decision-making constraints for a given medication can be defined using a number of constants:

- $t_{1/2}$ , its biological half life. This is the time taken for the plasma level of the medication to halve.
- $B$ , the typical dose amount consumed.
- $G$ , the amount of time one needs to leave between two doses.
- $m$ , the maximum number of doses within the planning horizon (e.g. 24 hours).

Given the half life, and the plasma level of a drug,  $D$ , the rate  $r$  at which the plasma concentration of the drug is decreasing can be calculated as:

$$r = D \cdot \frac{\ln 2}{t_{1/2}}$$

Taking the integral of this, the level of a drug at a time  $t$  ( $D(t)$ ) after some reference point ( $D_0$ ) can be written as:

$$D(t) = \frac{D_0}{2^{\frac{t}{t_{1/2}}}}$$

As we are planning the consumption of medication, we can think of the trajectory of drug levels during a solution plan as comprising time points at which medication is taken (when the drug level increases), interspersed with intervals in which the medication level reduces with rate  $r$ . More formally:

- At time  $t_0$ , the medication level takes some pre-defined value  $D_0$  (the initial plasma level).
- At subsequent ordered time points  $[t_1..t_{n-1}]$  an amount of medication  $[B_1..B_{n-1}]$  is taken. For this paper, we assume these values are either 0 or  $B$  – the dose that could in principle be consumed.
- At time  $t_n$ , no medication is taken – this represents the end of a finite horizon over which the plan must succeed.

With this representation, we can define the constraints on medication consumption. First, if a dose is taken at time  $t_i$ , the next dose cannot be taken at a time before  $t_i + G$ :

$$\forall_{i \in [1..n]} \left( B_i = 0 \vee \left( \forall_{j \in [i+1..n]} (B_j = 0 \vee t_j - t_i \geq G) \right) \right)$$

Second, the maximum number of doses within the planning horizon is  $m$ :

$$|\{i \cdot i \in [1..n-1] \wedge B_i \neq 0\}| \leq m$$

A medication plan can be said to be safe if it satisfies these two constraints. A separate matter is what therapeutic benefit is provided: taking no medication at all never exceeds the limits of the prescription, but of course is of no benefit to the patient. The medication level  $D(t_i)$  at time  $t_i$  ( $i > 0$ ) can be defined as:

$$D(t_i) = \frac{D(t_{i-1}) + B_{i-1}}{2^{\frac{t_i - t_{i-1}}{t_{1/2}}}}$$

The interaction between planning decisions and plasma levels arises when considering how a plan may constrain what are permissible drug levels at different points within the planning horizon.

In this paper we will focus on pain relief management with a single painkiller as an exemplar problem, so can discuss drug levels in terms of desired levels of pain relief ( $pr$ ).

In the simple case, the schedule of desired pain relief is static: at defined intervals throughout the day, the pain relief must be no lower than a minimum threshold. For instance, suppose a patient goes to work at 9am every morning and finishes work at 5pm every evening. We could then expect the minimum pain relief to be higher within this interval than at other times.

More persuasively, from a planning point of view, the desired pain relief is dynamic, depending on the actions used in the plan. For instance, during some actions (shopping for groceries, walking, and so on) a greater level of pain relief may be needed than at other times. A plan then must have a reasonable causal structure – as would be the case for planning a user’s day modulo medication requirements – but additionally, the actions may have preconditions referring to pain relief that must hold during their execution. To meet these, in turn, requires the use of actions that correspond to taking medication. The resulting plan then gives the patients a personalised schedule of times at which they should take their medication, along with a plan for the day for their other activities, to ensure they get the right pain relief at the right time.

### 4 Modelling Pharmacokinetics in PDDL+

Having now set out the mathematical model of medication levels that we will use, we now map this to PDDL+.

First, to model the pharmacokinetics, a process is used. This runs whenever there is a non-zero amount of medication in the bloodstream; and decreases the medication level at rate  $r$ . In our example of single-medication pain relief, the variable  $pr$  represents pain relief, and  $ke$  represents the elimination constant for the medication:

$$ke = \frac{\ln 2}{t_{1/2}}$$

The process can then be written as follows:

```
(:process decay
:parameters ()
:precondition
  (> (pr) 0)
:effect
  (decrease (pr) (* #t (* (pr) (ke))))
)
```

For clarity, the ‘decrease’ line can be read as:

$$-\frac{dpr}{dt} = pr \cdot ke$$

Alongside this process, there is one action that changes *pr*: **consume**. This is a durative action with duration *G* and has the following preconditions and effects:

- To start the action, a proposition *safe-to-consume* (true initially) must be true; and a variable *doses* (0 initially) must be less than *m* (the maximum number of doses).
- When started, *safe-to-consume* is deleted; *doses* is increased by 1; and the pain relief level *pr* is increased by *B* – the dose of medication.
- At the end of the action, *safe-to-consume* is added

Effectively, *safe-to-consume* and *doses* perform the requisite book-keeping to enforce the constraints on maximum medication consumption. *safe-to-consume* acts as a semaphore: no two consume actions can overlap; and the duration *G* serves to ensure the minimum specified time between doses is thereby respected. *doses* is a simple counter, to ensure that if a dose is to be taken, the maximum safe limit for the period over which we are planning cannot be exceeded.

Having defined the pharmacokinetics, and constrained medication taking, what is left is to define the minimum pain relief. For the case of a static minimum pain relief schedule, the minimum pain relief level *minpr* can be set using Timed Initial Fluents (Piacentini, Fox, and Long 2015) (TIF) – these specify the new value for *minpr* at each time it needs to change. For instance, taking plan time units to be minutes counting from midnight:

```
(= (minpr) 0)
(at 420 (= (minpr) 100))
(at 540 (= (minpr) 200))
(at 1020 (= (minpr) 100))
(at 1320 (= (minpr) 0))
```

...sets the minimum pain relief to 100 at time 420 (7:00), to 200 at time 540 (9:00), then back down to 100 and 0 later in the day.

Having set the schedule, to ensure at all times *pr* ≥ *minpr*, we use a PDDL+ event that fires at the first time this is not the case:

```
(:event prfailure
:parameters ()
:precondition (and
  (< (pr) (minpr))
  (min-check-passed))
)
:effect
  (not (min-check-passed))
)
```

The proposition *min-check-passed* is true in the initial state, required as a goal, and not added by any other action. Hence, if at any point in the plan the value of *pr* falls below *minpr*, the proposition is deleted, and a dead-end is reached: it is impossible to re-achieve this goal.

For dynamic pain relief levels, as set by actions, this is somewhat simpler: if an action requires some level of minimum pain relief during its execution, then this can be added as an *over all* condition, for instance:

```
(over all (>= (pr) 300))
```

...will ensure that the pain relief level is at least 300 during the execution of the action. The ‘TIF plus event’ model of a static pain relief schedule is unaffected, as this condition alone ensures that there is enough pain relief during the execution of the action.

The practical upshot of this PDDL+ encoding is that it allows a pharmacokinetic model to be specified as a background context into which a planning model can be specified.

## 5 Planning using a Linearise → Validate Cycle

In our initial experimentation with our PDDL+ model as specified thus far, a restricted range of planners were found that could reason with the negative-exponential numeric change induced by the process (e.g. UPMURPHI (Penna et al. 2009) and DINO (Piotrowski et al. 2015)); and some that could reason with processes, but only if the effects are linear (e.g. the extension of OPTIC described in (Coles and Coles 2014)). There is something of a trade-off between these two classes of planner. UPMURPHI *et al.* are capable of handling negative-exponentials and other non-linear domain features, and have been used in a number of applications including battery load management (Fox, Long, and Magazzeni 2012). Conversely, if linear change is sufficient, OPTIC’s heuristic forward-search approach is a good choice; but, assuming pharmacokinetics are linear is not reasonable<sup>1</sup>.

Desiring to maintain the benefits of using OPTIC, we present an iterative approach where a linear approximation is incrementally refined until the plans found are valid according to the non-linear domain model. This is related to the discretise–validate approach of UPMurphi, but instead of discretising time (notionally, on the X-axis), we linearise by segmenting the values of variables (on the Y-axis).

<sup>1</sup>Zero-order pharmacokinetics are rare; one notable exception is ethanol, but this has a narrow range of medical applications.

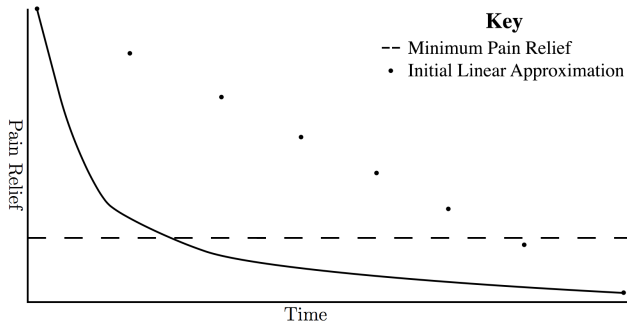


Figure 1: Initial Linear Approximation

As a starting point, we must devise an initial linear approximation. For this, we refer to:

- $lb$ , the lower bound on what is an interesting medication level. This cannot be 0, as mathematically, a negative exponentially decreasing drug plasma level will only reach 0 as  $t \rightarrow \infty$ . We instead use a nominal value of 1% of a dose (for paracetamol, a value of 10, i.e 10mg).
- $ub$ , the upper bound on what is an interesting medication level. This corresponds to taking the maximum number of doses  $m$  in succession, with each dose separated by the minimum time between doses,  $G$ .
- $\overline{r_{ub,lb}}$ , the average medication decay rate over the time taken for the plasma level to fall from  $ub$  to  $lb$ :

$$\overline{r_{ub,lb}} = \frac{ub - lb}{\frac{-t_{1/2}}{\ln 2} \cdot \ln(\frac{lb}{ub})}$$

This is depicted in Figure 1. The solid line shows the negative-exponential change in pain relief, assuming the pain relief at time 0 is  $ub$ . The x-axis ranges from 0 until the time at which the pain relief level would reach  $lb$ . The dotted line shows a linear approximation spanning this time, with gradient  $\overline{r_{ub,lb}}$ .

A substantial caveat is that the initial linear approximation substantially over-estimates pain relief. If there is a minimum pain relief threshold (e.g. the dashed line in Figure 1), then actual pain relief will fall below the threshold far sooner than would be considered to be the case according to the linear approximation. But, we do get a simple linear process, with a constant (linear) effect:

```
(:process decay_ub_lb
:parameters ()
:precondition
  (and (>= (pr) lb) (<= (pr) ub))
:effect
  (decrease (pr) (* #t r_ub_lb))
)
```

Solving this linearised problem, then validating the plan against the non-linear model using VAL (Howey, Long, and Fox 2004), will identify where the inaccuracies inherent in the linearisation have caused issues. This is evidenced in one of two ways:

---

#### Algorithm 1: Linearise–Validate Cycle

---

**Data:**  $P$ , the non-linear planning domain and problem;  
 $bounds = \{ub, lb\}$ , the initial bounds

**Result:** A solution plan,  $\Pi$

```
1  $P_{base} \leftarrow P$ , with the decay process removed;
2 while true do
3    $sorted \leftarrow [bounds, \text{sorted in descending order}]$ ;
4    $P' \leftarrow P_{base}$ , with a linear process for each
     successive pair in  $sorted$ ;
5    $\Pi \leftarrow \text{solve } P' \text{ using planner}$ ;
6   if  $\Pi$  is a valid solution to  $P$  then return  $\Pi$ ;
7   foreach  $pr \in \text{VAL's diagnostic trace for } \Pi \text{ using}$ 
      $\text{model } P$  do
8      $bounds \leftarrow bounds \cup \{pr\}$ 
```

---

- The event `prfailure` occurred, deleting (`min-check-passed`) – as this is a goal, the solution plan is invalid.
- A precondition on an action referring to  $pr$  was unsatisfied at some time – and hence the solution plan is invalid.

In both of these cases, VAL produces a diagnostic trace: a time-stamped progression through the plan, including what the value of  $pr$  was at each happening in the plan, as evaluated against the non-linear domain. With this information, we can refine the linearisation: instead of having a single-segment linear process spanning the whole range  $ub$  to  $lb$ , we can have several processes each covering one segment of this range.

Our motivation for refining the linearisation to give the right value of  $pr$  at happenings is based on the observation that error in the linearisation is acceptable, so long it gives the right value when it matters; i.e. when `prfailure` would fire, or a precondition referring to  $pr$  would be violated. Hence, we ensure that on each iteration, the plan found with the previous linearisation will not be admitted by the new linearisation. This does not guarantee that a solution to the non-linear model will be found on the second iteration, but it does mean the model is iteratively refined to exclude apparently attractive but actually infeasible solutions.

Our approach is shown in Algorithm 1. We begin by running the planner to find a solution based on the initial approximation; i.e. starting with the initial values of  $ub$  and  $lb$ . Hence, at line 4, when the linearised model  $P'$  is generated, we have only one pair of bounds in the list, and generate a single process covering this range, with an effect with gradient  $\overline{r_{ub,lb}}$ . A solution  $\Pi$  to  $P'$  is then found.

As noted earlier, it is likely that when using the initial linearisation,  $\Pi$  will not be a solution to  $P$  – as  $P'$  overestimates the actual pain relief throughout the day. Hence, to refine the linearisation, we refer to the happenings in the diagnostic trace from VAL, an example of which can be seen in Figure 2, and keep all calculated values of  $pr$  (marked in boldface); i.e. for the plan  $\Pi$ , what values of  $pr$  were seen according to the model  $P$ . Each of these is added to the set of bounds (line 8). With this updated set of bounds, the loop starts again, generating an updated linear problem  $P'$ ,

```

Checking next happening (time 240)
Updating (pr) (1300) by 515.905 assignment

Checking next happening (time 240)
Adding (safe-to-consume)

Checking next happening (time 486.079)
Updating (pr) (515.905) by 200 assignment

EVENT triggered at (time 486.079)
Triggered event (prfailure)
Deleting (min-check-passed)

Checking next happening (time 544.824)
Updating (pr) (200) by 159.509 assignment

...

Checking next happening (time 784.824)
Updating (pr) (1159.51) by 460.152 assignment

```

Figure 2: Example Stack Trace from VAL

and once again attempting to find a solution plan. For the updated problem, the bounds are sorted, and a process generated for each adjacent pair of bounds, each with its own  $\overline{r_{ub,lb}}$  value.

## 6 Evaluation

As an initial evaluation of our approach, we generated a sequence of problems with increasing planning horizon and a fixed  $minpr$  level, thereby necessitating increased number of doses as problem sizes increases. The planning horizons tested started at 540 minutes (i.e. 9 hours), and increased by 60 minutes each time.

The reference drug used was paracetamol, with the following initial parameters:

- $t_{1/2}$  (half life): 180 minutes
- $B$  (dose amount): 1000mg
- $G$  (gap between doses): 240 minutes
- $m$  (max doses): 4 doses
- $lb$ , (lower bound): 10mg
- $ub$ , (upper bound): 3000mg
- $minpr$ : 200mg.

Our linearise-validate cycle was implemented in a planner-independent way, but the only candidate planner that yielded solutions was the extension of OPTIC to support linear PDDL+ (Coles and Coles 2014). A number of other planners were considered (Cashmore et al. 2016; Piotrowski et al. 2015; Penna et al. 2009), but publicly available implementations of planners were unable to solve problems (non-linear and linear). The results for this configuration are shown in Table 1. All problems were solved in at most two iterations, with the time for the first iteration shown in the row  $t_{initial}$ , and for the refined iteration in the row  $t_{refined}$ .

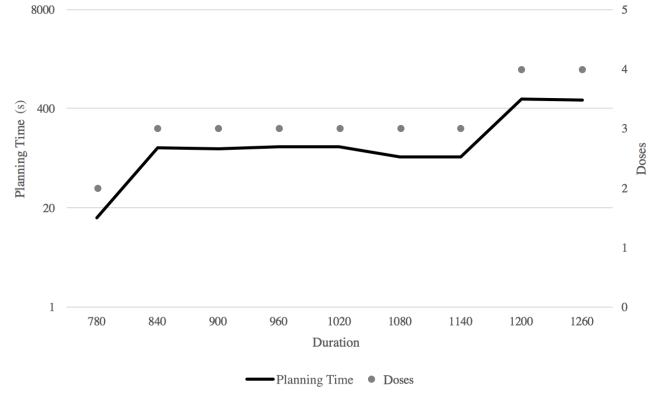


Figure 3: Comparison Between Doses Needed and Planning Time (in seconds)

For shorter planning horizons, we see some interesting results.

- For a horizon of 540 minutes, the solution for the initial linearisation used only a single dose of medication; the resulting solution did not validate, so the linear model was refined, and a new plan found.
- For horizons in the range 600–720, the solution for the initial linearisation recognised the need to take two doses of medication; and these were taken back-to-back. The resulting plan was incidentally a valid solution for the non-linear model.
- For a horizon of 780, two doses were needed (as with 600–720) but the timing of the doses is more important: the second dose needs to be delayed to come somewhat more than four hours after the first. This was not recognised by the initial linearisation (which over-estimated pain relief at time 780); but was compensated for in the refined linearisation by appropriately delaying the second dose.

For the larger problems, the initial linearisation was never valid. The planning time for the refined model was strongly correlated with the number of doses to be taken – this is shown more clearly in Figure 3, where the planning time (left Y-axis) tracks the number of doses needed (right Y-axis).

To gain further insights into the linearisation process, we looked at the solution plan found for the longest planning horizon (1260), after the first and second iteration. These plans were validated against three models:

- The refined linearisation
- The non-linear model (i.e. exponential decay)
- The initial linearisation

Figure 4 shows the calculated value of  $pr$  for the plan after the first iteration, validated against each of these. The first plan took three doses back-to-back. With reference to the initial linearisation (the dashed line), this is reasonable – a fourth dose was not necessary to complete the plan. As can be seen, the initial approximation is as expected extremely

	Planning Horizon (minutes)												
	540	600	660	720	780	840	900	960	1020	1080	1140	1200	1260
$t_{\text{initial}}$	0.05	0.03	0.03	0.04	0.03	0.04	0.03	0.03	0.06	0.06	0.06	0.09	0.09
$t_{\text{refined}}$	0.71	-	-	-	11.72	164.19	162.99	161.29	160.09	110.05	111.2	654.77	649.86
<b>total</b>	0.76	0.03	0.03	0.04	11.75	164.23	163.04	161.32	160.12	110.11	111.26	654.83	649.95
<b>doses</b>	2	2	2	2	2	3	3	3	3	3	3	4	4

Table 1: Planning times (seconds) for fixed  $\text{minpr}$ , increasing plan horizon

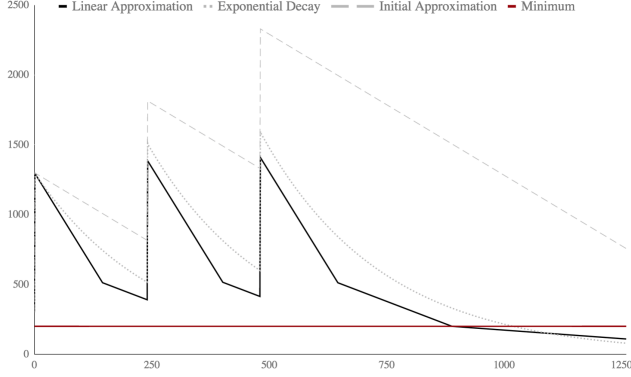


Figure 4: Pain Relief Levels with Three Models, using the Initial Approximation Plan

$ub$	3000.0	515.905	515.901	200.0
$lb$	515.905	515.901	200.0	10.0
$\bar{r}_{ub,lb}$	5.43373	1.986648	1.283748	0.2442325

Table 2: Example bounds after one iteration (horizon 1260)

optimistic, with the calculated value of  $pr$  significantly exceeding the actual negative-exponential value (the dotted line). The refined linearisation (solid line) avoids falling into the same trap: a three-dose solution would cross the  $\text{minpr}$  threshold, so would never be returned as a solution by the planner.

An analogous graph for the plan after the second iteration, found by the planner using the refined linearisation, is shown in Figure 5. Crucially, a fourth dose is now taken. In particular, the planner scheduled doses to be taken at the earliest possible instance for the first three doses (at time 0, 240 and 480) and waited until the latest possible time to take the fourth dose; i.e. medication was consumed just before the  $\text{minpr}$  threshold was crossed, the point at which the  $\text{prfailure}$  event would otherwise have fired.

For reference, with a horizon of 1260, five bounds were used: the initial bounds of 3000 and 10, and a further three in between. This yielded four linear processes, whose parameters are shown in Table 2.

To test whether the planner could scale over a horizon beyond 1260 minutes, we created a problem with a horizon of 1500 minutes (25 hours), using a Timed Initial Literal (Hoffmann and Edelkamp 2005) to mark the point at which the day changed (to limit doses consumed per 24 hours). Al-

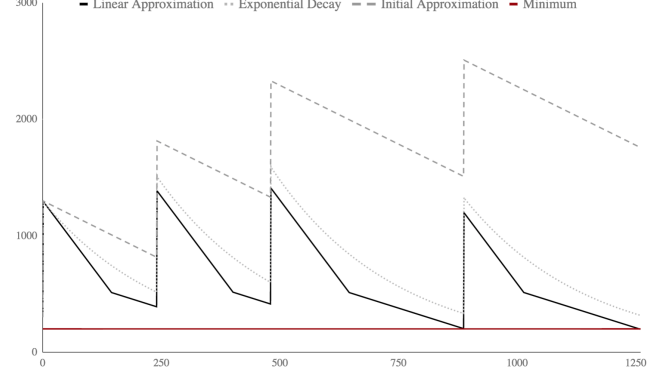


Figure 5: Pain Relief Levels with Three Models, using the Refined Approximation Plan

though the planner was able to find a solution, needing five doses, it took almost two hours – adding the Timed Initial Literal to switch from one day to the next had a substantial effect on the size of the search space.

## 7 Discussion and Future Work

With our linearise-validate approach, we have shown we can solve problems in this domain, and have presented an evaluation to show its efficacy on fixed-minimum-pain-relief tasks. The evaluation problem as it stands could be seen as a scheduling problem rather than a planning problem, but our motivation for doing this within PDDL+ is to allow task and activity planning to take place in the context of medication scheduling.

We will now briefly discuss some of our future research directions.

### 7.1 Polymedicine

Our evaluation here considered only a single drug; the next step is to look at polypharmacy. As almost a quarter of the UK population are on at least three prescriptions (Scholes, Faulding, and Mindell 2014), this is a substantial area of interest.

In the case of painkillers, these are often complementary. If a patient was only on one drug, it may be difficult to give adequate pain relief due to the constraints of the drug itself. For example, paracetamol is an effective painkiller, but the dosage restrictions mean it is a challenge to use it as a monotherapy to give sufficient pain relief for a patient's routine. Thus, if we take into account multiple painkillers (for

example paracetamol and ibuprofen), a combined schedule of the two gives greater potential for pain management (taking both at the same time), and greater flexibility (taking them at different times). The modifications to the model to support this are relatively straightforward: rather than using a single *pr* variable, use multiple such variables (one for each painkiller) and define conditions on pain relief to refer to a weighted sum of these.

A more challenging case is where there are interactions between medication. Ideally, two adversely interacting medications would not be taken concomitantly, but it is sometimes unavoidable. To handle pharmacokinetic interactions (one drug affects the rate of metabolism of another), the drug decay processes would need to be updated. How to do this well remains an open challenge.

## 7.2 Planning to avoid side effects

As discussed earlier, activity-specific drug plasma level requirements can be incorporated into the preconditions of actions. This provides a mechanism for allocating doses of medication around a patient's daily routine.

A further consideration is the side-effects of medication, as well as their desired effects. These side-effects place various constraints on how medication should be taken. For instance, some medication require activities to take place before or after consumption. For example:

- Ibuprofen cannot be taken on an empty stomach, or it will cause irritation, so an ideal plan would include meal-times as well as medication times.
- Codeine causes drowsiness, so patients should avoid taking it before driving or operating machinery.
- Paracetamol conversely has no such constraints, but the daily maximum dose is quite limited.

Considering a full plan of action for the day for a patient, covering a wide range of their daily activities, there is good scope to improve the management of their medication to reduce adverse effects.

## 7.3 Plan quality metrics

Thus far, our discussion has been on finding plans that meet hard constraints, in terms of drug plasma levels during times of the day, or during activities. We could hope to improve a patient's quality of life further by finding plans that are of good quality.

The question then is what the quality metric should be. In the context of pain relief, whilst a minimum pain relief may be specified, the patient may for the sake of comfort wish to avoid their pain relief getting quite down to this minimum. A good candidate for a plan quality metric is to maximise the minimum gap between *pr* and *minpr* seen during the plan. With reference to Figure 5, this would have the effect of delaying the second/third doses to avoid *pr* going quite so low before the fourth dose was taken: the same medication is taken, but the plan is subjectively better.

The caveat here though is that whilst 'max of min' or 'min of max' metrics are common in various scheduling problems (e.g. Job-Shop Scheduling) they are non-standard in

planning, and cannot be elegantly expressed in PDDL+. As OPTIC is already using a MIP to check that the preconditions on solution plans hold, it already has the framework to use more powerful plan metrics, by setting these as the MIP objective function; we will be exploring this in our future work.

## 8 Conclusions

In this paper, we presented a PDDL+ model of drug pharmacokinetics, to provide a context in which to find solution plans that consider, *inter alia*, the consumption of medication. As heuristic forward-search planners do well in terms of causal reasoning, but often handle only linear dynamics, we devised a linearise-validate approach for solving these problems, by iteratively refining a linear approximation of the domain using the diagnostic trace returned by the plan validator, VAL.

An initial evaluation, implemented as a wrapper around the planner OPTIC, demonstrates the feasibility of this approach. The results are exciting, opening up the opportunity for future work both in terms of more comprehensive model development, and more generally improving the range of quality metrics that can be handled by PDDL-based planners in hybrid domains.

## References

- Benton, J.; Coles, A. J.; and Coles, A. I. 2012. Temporal planning with preferences and time-dependent continuous costs. In *Proceedings of the Twenty Second International Conference on Automated Planning and Scheduling (ICAPS)*.
- Cashmore, M.; Fox, M.; Long, D.; and Magazzeni, D. 2016. A Compilation of the Full PDDL+ Language into SMT. In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling (ICAPS)*.
- Coles, A. J., and Coles, A. I. 2014. PDDL+ Planning with Events and Linear Processes. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS)*.
- Fox, M., and Long, D. 2003. PDDL2.1 : An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*.
- Fox, M., and Long, D. 2006. Modelling Mixed Discrete-Continuous Domains for Planning. *Journal of Artificial Intelligence Research*.
- Fox, M.; Long, D.; and Magazzeni, D. 2012. Plan-based policies for efficient multiple battery load management. *Journal of Artificial Intelligence Research*.
- Geenen, S.; Yates, J. W. T.; Kenna, J. G.; Bois, F. Y.; Wilson, I. D.; and Westerhoff, H. V. 2013. Multiscale modelling approach combining a kinetic model of glutathione metabolism with pbpk models of paracetamol and the potential glutathione-depletion biomarkers ophthalmic acid and 5-oxoproline in humans and rats. *Integrative Biology* 5:877–888.



Hirshfield, L.; Giridhar, A.; Taylor, L.; Harris, M.; and Reklaitis, G. 2014. Dropwise additive manufacturing of pharmaceutical products for solvent-based dosage forms. *Journal of Pharmaceutical Sciences* 103:496–506.

Hoffmann, J., and Edelkamp, S. 2005. The Deterministic Part of IPC-4: An Overview. *Journal of Artificial Intelligence Research*.

Howey, R.; Long, D.; and Fox, M. 2004. VAL: Automatic Plan Validation, Continuous Effects and Mixed Initiative Planning using PDDL. In *The 16th IEEE International Conference on Tools with Artificial Intelligence*, 294–301.

Penna, G. D.; Magazzeni, D.; Mercurio, F.; and Intrigila, B. 2009. UPMurphi: A Tool for Universal Planning on PDDL+ Problems. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS)*.

Piacentini, C.; Fox, M.; and Long, D. 2015. Planning with numeric timed initial fluents. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*.

Piotrowski, W.; Fox, M.; Long, D.; Magazzeni, D.; and Mercurio, F. 2015. Heuristic Planning for Hybrid Systems. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*.

Sadee, W., and Dai, Z. 2005. Pharmacogenetics/genomics and personalized medicine. *Human Molecular Genetics*.

Scholes, S.; Faulding, S.; and Mindell, J. 2014. Use of prescribed medicines. In *Health Survey for England – 2013*. chapter 10.

The Academy of Medical Sciences. 2014. Patient Adherence to Medicines.